



**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОУ ВПО НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Р.Е. АЛЕКСЕЕВА**

**ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

Кафедра "Вычислительные системы и технологии"

ПРОГРАММИРОВАНИЕ

**Отчёты
о выполнении лабораторного практикума**

Выполнил студент группы 22-ИВТ-2 Лисенков Кирилл
Алексеевич

_____ «__» _____ 20__ г.
(личная подпись) (дата)

Провел старший преподаватель кафедры
«Вычислительные системы и технологии»
Мартынов Дмитрий Сергеевич

_____ «__» _____ 20__ г.
(личная подпись) (дата)

НИЖНИЙ НОВГОРОД 2023

Создать пользовательский класс TCharArray (массив), используемый для хранения элементов типа char. В данном классе должен быть реализован метод at для доступа к элементу символьного массива с проверкой корректности значения индекса элемента массива. Кроме того, необходимо перегрузить операцию [] для доступа к элементам массива. Используя класс TCharArray в качестве родительского, создать производный от него пользовательский класс String, используемый для хранения символьных строк. Для данного класса перегрузить следующие операции: '+', '==', '>', 'Разрабатываемая программа предназначена для хранения массива экземпляров класса. Перечень атрибут класса (членов-данных) определяется исходя из задания во второй лабораторной работе. Созданная программа должна поддерживать управление на уровне аргументов командной строки (аргументов запуска)

```
#ifndef LIB_H
#define LIB_H
#include <cstdint>
#include <iostream>
#include <fstream>
class TCharArray {
protected:
    char* buf;
    size_t len;
public:
    TCharArray();
    TCharArray(char*);
    TCharArray(TCharArray const&);
    TCharArray(size_t);
    ~TCharArray();
    void set(char*);
    const char* get() const;
    char& operator[] (int);
    char& at(int);
};

class String : public TCharArray
{
public:
    String();
    String(char*);
    String(String const&);
    String(size_t);
    void set(char *);
    const char* get() const;
    String operator+(const String&);
    bool operator==(const String&);
    bool operator>(const String&);
    bool operator<(const String&);
    bool operator!=(const String&);
    friend std::istream& operator>>(std::istream&, String&);
};

class Travel{
private:
    String name;
    String country;
```

```

    double price;
    double duration;
    String transport;
public:
    void setName(const char* newName);
    void setCountry(const char* newCountry);
    void setPrice(const double newPrice);
    void setDuration(const double newDuration);
    void setTransport(const char* newTransport);
    const char* getName()const;
    const char* getCountry()const;
    const double getPrice()const;
    const double getDuration()const;
    const char* getTransport()const;
    friend std::istream& operator>>(std::istream&, Travel&);
    friend std::ostream& operator<<(std::ostream&, const Travel&);
    void infile(std::ofstream& file);
};
#endif

```

```

#include "lib.h"
#include <cstring>
#include <stdexcept>
#include <iostream>
#include <iomanip>
#include <fstream>

TCharArray::TCharArray(){
    buf = nullptr;
    len = 0;
}

TCharArray::TCharArray(char* s){
    len = std::strlen(s);
    buf = new char[len+1];
    std::strcpy(buf,s);
}

TCharArray::TCharArray(TCharArray const& other){
    len = other.len;
    buf = new char[len+1];
    std::strcpy(buf, other.buf);
}

TCharArray::TCharArray(size_t size){
    len = size;
    buf = new char[len+1];
}

TCharArray::~TCharArray(){
    delete[] buf;
}

void TCharArray::set(char* s){
    if(buf)
        delete[] buf;

```

```

    len = std::strlen(s);
    buf = new char[len+1];
    std::strcpy(buf,s);
    return;
}

const char* TCharArray::get() const{
    return buf;
}

char& TCharArray::operator[](int i){
    return buf[i];
}

char& TCharArray::at(int i){
    if ( (i > len+1) || (i < 0)){
        throw std::out_of_range("TCharArray Index out of range.");
    }
    return buf[i];
}

String::String() : TCharArray({})

String::String(char* s) : TCharArray(s){}

String::String(String const& other) : TCharArray(other){}

String::String(size_t size) : TCharArray(size){}

void String::set(char* s){
    TCharArray::set(s);
    return;
}

const char* String::get() const{
    return TCharArray::get();
}

String String::operator+(const String& other){
    String result(len+other.len);
    for(int i{0}; i < len; i++)
        result[i] = buf[i];
    for(int i{(int)len}; i < len+other.len; i++)
        result[i] = other.buf[i-len];
    return result;
}

bool String::operator==(const String& other){
    if(len != other.len)
        return false;
    for(int i{0}; i < len; i++){
        if(buf[i] != other.buf[i])
            return false;
    }
    return true;
}

bool String::operator>(const String& other){

```

```

        if(len < other.len)
            return false;
        return true;
    }
    bool String::operator<(const String& other){
        if(len > other.len)
            return false;
        return true;
    }
    bool String::operator!=(const String& other){
        if(len != other.len)
            return true;
        for(int i{0}; i < len; i++){
            if(buf[i] != other.buf[i])
                return true;
        }
        return false;
    }
}

std::istream& operator>>(std::istream& is, String& str){
    char buffer[100];
    is.getline(buffer,100);
    str.set(buffer);
    return is;
}

const char* Travel::getName()const{return name.get();}

const char* Travel::getCountry()const{return country.get();}

const double Travel::getPrice()const {return price;};

const double Travel::getDuration()const {return duration;};

const char* Travel::getTransport()const{return transport.get();}

std::istream& operator>>(std::istream& is, Travel& Travel){
    static int counter = 1;
    std::cout << "Запись: " << counter << ":\n";
    std::cout << "    Наименование: ";
    if(!(is >> Travel.name)){
        throw std::runtime_error("Ошибка Ввода");
    }
    std::cout << "    Страна: ";
    if(!(is >> Travel.country)){
        throw std::runtime_error("Ошибка Ввода");
    }
    std::cout << "    Стоимость: ";
    if(!(is >> Travel.price)){
        throw std::runtime_error("Ошибка Ввода");
    }
    std::cout << "    Продолжительность: ";
    if(!(is >> Travel.duration)){
        throw std::runtime_error("Ошибка Ввода");
    }
    std::cout << "    Вид транспорта: ";

```

```

    is >> Travel.transport;
    if(!(is >> Travel.transport)){
        throw std::runtime_error("Ошибка Ввода");
    }
    counter++;
    return is;
}

std::ostream& operator<<(std::ostream& os, const Travel& Travel){
    static bool first = true;
    if (first){
        os << "-----\n" << std::left << std::setw(20) << "|Name" << std::setw(20) << "|Country" << std::setw(10) <<
        "|Price" << std::setw(10) << "|Duration" << std::setw(20) << "|Transport|" << std::endl << "----\n";
        first = false;
    }
    os << std::left << std::setw(20) << Travel.getName() << std::left << std::setw(20) <<
    Travel.getCountry() << std::setw(10) << std::fixed << std::setprecision(2) << Travel.getPrice()
    << std::setw(10) << std::fixed << std::setprecision(2) << Travel.getDuration() << std::setw(20)
    << Travel.getTransport() << '\n';
    return os;
}

// figure: task1
//*****
// Filename: main.cpp
// Abstract: Простая программа на C++
// Description:
// Create Date: 2023 / 05 / 01
// Author: Студент ИРИТ Лисенков К.А. 22-ИБТ-2
// Notes / Platform / Copyright IRIT NNTU/ UNIX/Linux / FreeWare
//*****
#include <iostream>
#include <cstring>
#include <fstream>
#include <iomanip>
#include "lib.h"
int main(int argc, char** argv){
    std::cout << "*****\n" //Приветствие.
    << "* Нижегородский государственный технический университет *\n"
    << "* Лабораторная работа 4. Задание 1. Вариант 19 *\n"
    << "* Выполнил студент группы 22-ИБТ-2. Лисенков Кирилл *\n"
    << "*****\n";
    if((argc == 2) && ((strcmp(argv[1], "-h") == 0) || (strcmp(argv[1], "--help") == 0))){
//Получения справки
        std::cout << "Справка:\n"
        << "Добро пожаловать в программу для для хранения и обработки массива
переменных структурного типа данных.\n"
        << " -c [N] [file_name] - запуск программы в режиме создания электронной
таблицы записей, N – количество записей, file_name – имя текстового файла, в котором будет
сохранен массив (таблица) записей.\n"
        << " -r [N] [file_name] - запуск программы в режиме чтения содержимого
текстового файла file_name, на экран должны быть выведены не более N записей.\n";
        exit(0);
    }
}

```

```

else if((argc == 4) && ((strcmp(argv[1], "-c") == 0) || (strcmp(argv[2], "--create") == 0))){
    if(atoi(argv[2]) < 1){
        std::cout << "Ошибка, число записей не может быть отрицательным или равным нулю!";
        exit(-1);
    }
    size_t n = atoi(argv[2]);
    Travel* table = new Travel[n];
    std::ofstream file(argv[3]);
    std::ofstream &rfile = file;
    for(int i{0}; i < n; i++){
        std::cin >> table[i];
        file << table[i];
    }
    std::cout << "-----
\n" << std::left << std::setw(20) << "|Name" << std::setw(20) << "|Country" << std::setw(10) <<
"|Price" << std::setw(10) << "|Duration" << std::setw(20) << "|Transport|" << std::endl << "----
-----\n";

    for(int i{0}; i < n; i++){
        std::cout << table[i];
    }
}
else if((argc == 4) && ((strcmp(argv[1], "-r") == 0) || (strcmp(argv[2], "--read") == 0))){
//Чтения бд
    if(atoi(argv[2]) < 1){
        std::cout << "Ошибка, число записей не может быть отрицательным или равным
нулю!\n";
        exit(-1);
    }
    size_t n = atoi(argv[2]);
    std::ifstream file(argv[3]);
    if(!file.is_open()){
        std::cout << "Ошибка открытия файла!";
        exit(-1);
    }
    std::string line;
    int i{0};
    while (std::getline(file, line)){
        std::cout << line << std::endl;
        i++;
        if(i == n+4)
            break;
    }
}
else
    std::cout << "Некорректные аргументы командной строки. Укажите -h или --help для
получения справки\n";
    return 0;
}

```

Алгоритм "Хранение и обработка массива переменных структурного типа данных"

Начало

- | Выводим приветствие
- | Если аргументы командной строки равны "-h" или "--help"
- || Выводим справку по использованию программы
- || Завершаем программу

```

| Конец условия для получения справки
| Если аргументы командной строки равны "-с" или "--create"
| | Если второй аргумент N меньше 1
| | | Выводим сообщение об ошибке "Ошибка, число записей не может быть отрицательным или равным нулю!"
| | | Завершаем программу с кодом -1
| | Конец условия для проверки N
| | Преобразуем третий аргумент в тип size_t и присваиваем его значение переменной n
| | Создаем динамический массив table размером n типа Travel
| | Открываем файл с именем, указанным в четвертом аргументе, для записи и присваиваем его ссылке переменной
file
| | Если не удалось открыть файл
| | | Выводим сообщение об ошибке "Ошибка открытия файла!"
| | | Завершаем программу с кодом -1
| | Конец условия для проверки открытия файла
| | Цикл для каждого i от 0 до n-1
| | | Вводим данные с клавиатуры и сохраняем их в table[i]
| | | Записываем данные в файл
| | Конец цикла для i
| | Выводим заголовок таблицы
| | Цикл для каждого i от 0 до n-1
| | | Выводим данные из table[i]
| | Конец цикла для i
| Конец условия для создания таблицы записей
| Если аргументы командной строки равны "-г" или "--read"
| | Если второй аргумент N меньше 1
| | | Выводим сообщение об ошибке "Ошибка, число записей не может быть отрицательным или равным нулю!"
| | | Завершаем программу с кодом -1
| | Конец условия для проверки N
| | Преобразуем третий аргумент в тип size_t и присваиваем его значение переменной n
| | Открываем файл с именем, указанным в четвертом аргументе, для чтения и присваиваем его ссылке переменной
file
| | Если не удалось открыть файл
| | | Выводим сообщение об ошибке "Ошибка открытия файла!"
| | | Завершаем программу с кодом -1
| | Конец условия для проверки открытия файла
| | Читаем строки из файла и выводим на экран не более N записей
| Конец условия для чтения файла
| Если не выполнилось ни одно из условий выше
| | Выводим сообщение об ошибке "Некорректные аргументы командной строки. Укажите -h или --help для
получения справки"
Конец программы

```

```

*****
Нижгородский государственный технический университет *
Лабораторная работа 4. Задание 1. Вариант 19 *
Выполнил студент группы 22-ИВТ-2: Лисенков Кирилл *
*****
Справка:
добро пожаловать в программу для хранения и обработки массива переменных структурного типа данных.
-с [N] [file_name] - запуск программы в режиме создания электронной таблицы записей, N - количество записей, file_name - имя текстового файла, в котором будет сохранен массив (таблица) записей.
-г [N] [file_name] - запуск программы в режиме чтения содержимого текстового файла file_name, на экран должны быть выведены не более N записей.

```



```
~/Документы/University/2 semester/Programming/Labo4 ./App1 -c 2 file.txt
```

```
*****
* Нижегородский государственный технический университет *
* Лабораторная работа 4. Задание 1. Вариант 19          *
* Выполнил студент группы 22-ИВТ-2. Лисенков Кирилл      *
*****
```

Запись: 1:

Наименование: a
Страна: a
Стоимость: 1
Продолжительность: 2
Вид транспорта: a

Запись: 2:

Наименование: b
Страна: b
Стоимость: 3
Продолжительность: 4
Вид транспорта: b

Name	Country	Price	Duration	Transport
a	a	1.00	2.00	a
b	b	3.00	4.00	b

```
~/Документы/University/2 semester/Programming/Labo4 ./App1 -r 1 file.txt
```

```
*****
* Нижегородский государственный технический университет *
* Лабораторная работа 4. Задание 1. Вариант 19          *
* Выполнил студент группы 22-ИВТ-2. Лисенков Кирилл      *
*****
```

Name	Country	Price	Duration	Transport
a	a	1.00	2.00	a